

**Nama: Hilmi Hanif**  
**NIM: A11.2021.13491**  
**Kelompok: 4211**

## **Latihan Procedure Trigger**

### **Create table**

#### **Table barang**

```
create table barang (  
  kode_brg char(3),  
  nama_brg varchar(25),  
  satuan varchar(10),  
  jml_stok int(5),  
  primary key(kode_brg)  
)engine=InnoDB;
```

#### **Table Jual**

```
create table jual (  
  kode_jual char(2),  
  nama_kons varchar(25),  
  alamat_kons text,  
  kota_kons varchar(25),  
  tgl_jual date,  
  jml_jual int(4),  
  kode_brg char(3),  
  primary key(kode_jual)  
)engine=InnoDB;
```

#### **Table Beli**

```
create table beli (  
  kode_beli char(2),  
  nama_supplier varchar(25),  
  tgl_beli date,  
  jml_beli int(4),  
  kode_brg char(3),  
  primary key(kode_beli)  
)engine=InnoDB;
```

## Create procedure

### Insert Data Jual

Delimiter \$\$

```
create procedure insertDataJual (  
    xkode_jual char(2),  
    xnama_kons varchar(25),  
    xalamat_kons text,  
    xkota_kons varchar(25),  
    xtgl_jual date,  
    xjml_jual int(4),  
    xkode_brg char(3)  
)  
begin  
    insert into jual values(  
        xkode_jual,  
        xnama_kons,  
        xalamat_kons,  
        xkota_kons,  
        xtgl_jual,  
        xjml_jual,  
        xkode_brg  
    );  
    select * from jual;
```

end \$\$

Delimiter ;

### Insert Data Beli

delimiter \$\$

```
create procedure insertDataBeli (  
    xkode_beli char(2),  
    xnama_supplier varchar(25),  
    xtgl_beli date,  
    xjml_beli int(4),  
    xkode_brg char(3)  
)  
begin  
    insert into beli  
    value (  
        xkode_beli,  
        xnama_supplier,
```

```

        xtgl_beli,
        xjml_beli,
        xkode_brg
    );
    select * from beli;
end$$
delimiter ;

```

### **create trigger**

#### **Insert Jual**

```

delimiter $$
create trigger insertJual
    after insert
    on jual
    for each row
begin
    update barang
    set jml_stok = jml_stok - new.jml_jual
    where kode_brg=new.kode_brg;
end$$
delimiter ;

```

#### **Insert Beli**

```

delimiter $$
create trigger insertBeli
    after insert
    on beli
    for each row
begin
    update barang
    set jml_stok = jml_stok + new.jml_beli
    where kode_brg=new.kode_brg;
end$$
delimiter :

```

## Delete Jual

delimiter \$\$

create trigger deleteJual

after delete

on jual

for each row

begin

update barang

set jml\_stok = jml\_stok + old.jml\_jual

where kode\_brg = old.kode\_brg;

end\$\$

delimiter ;

## Delete Beli

delimiter \$\$

create trigger deleteBeli

after delete

on beli

for each row

begin

update barang

set jml\_stok = jml\_stok - old.jml\_beli

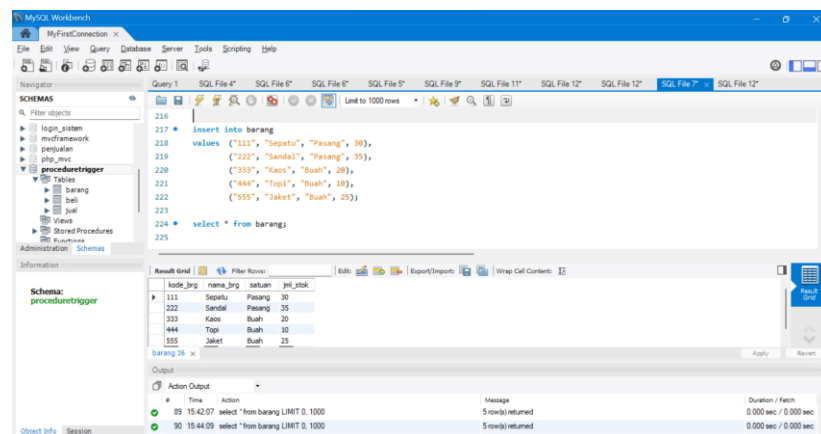
where kode\_brg=old.kode\_brg;

end\$\$

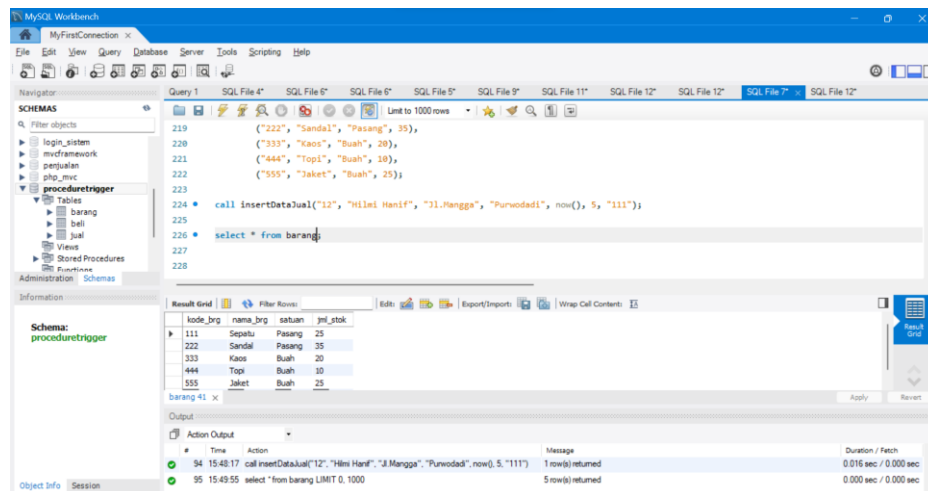
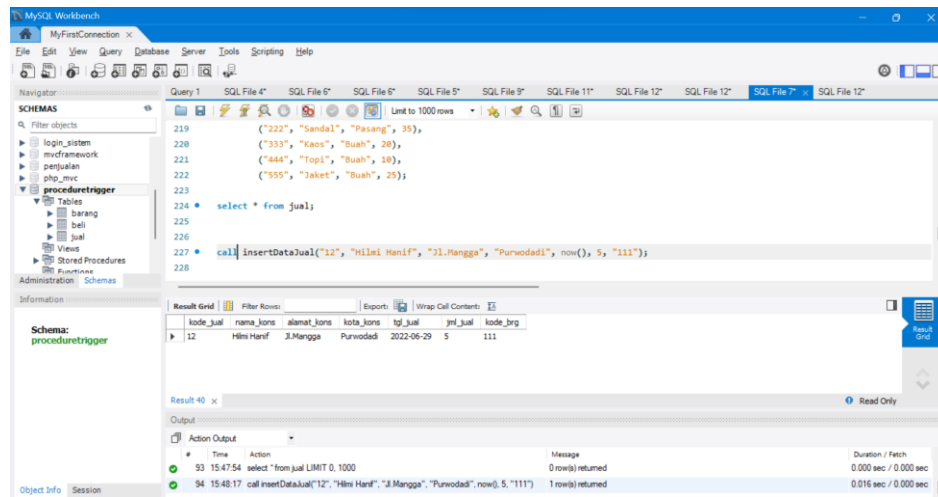
delimiter :

## Hasil

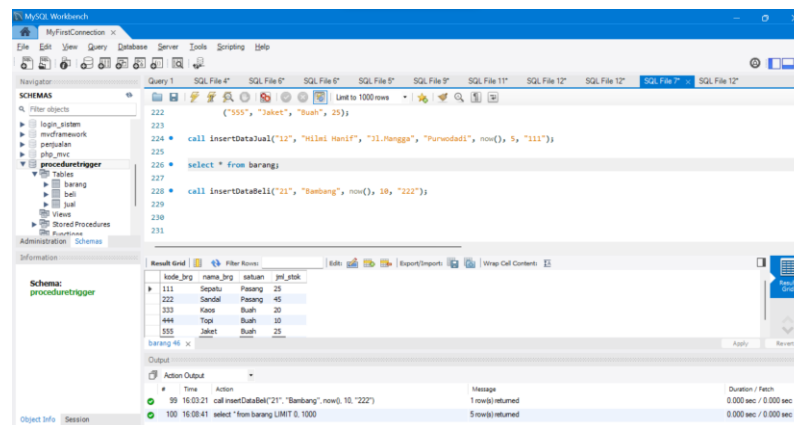
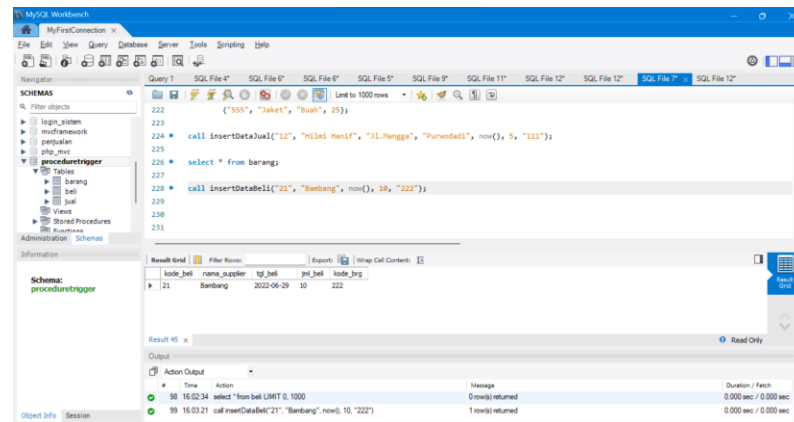
## Memasukan Data pada tabel barang



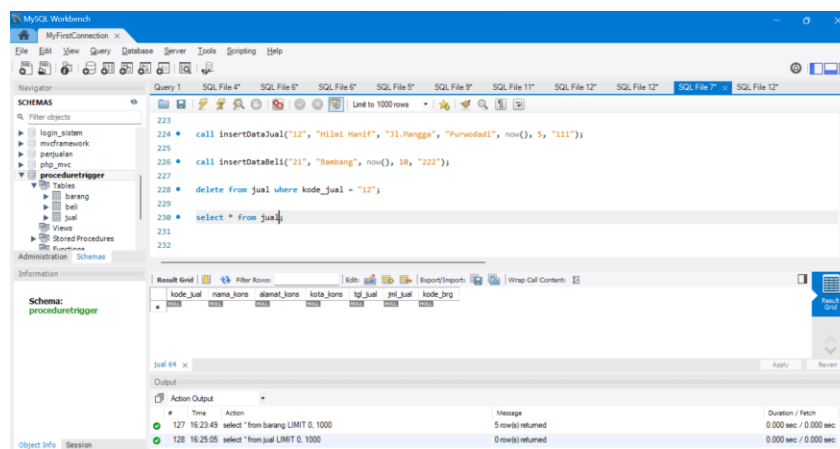
Memasukan Data Jual menggunakan store Procedure, Jika data jual dimasukan dengan jumlah jual tertentu maka data tabel barang yang memiliki hubungan kode barang maka akan di Trigger dan data pada tabel barang akan di kurangi. Jadi disini jumlah barang pada kode\_brg = 111 akan di kurangi menjadi 25

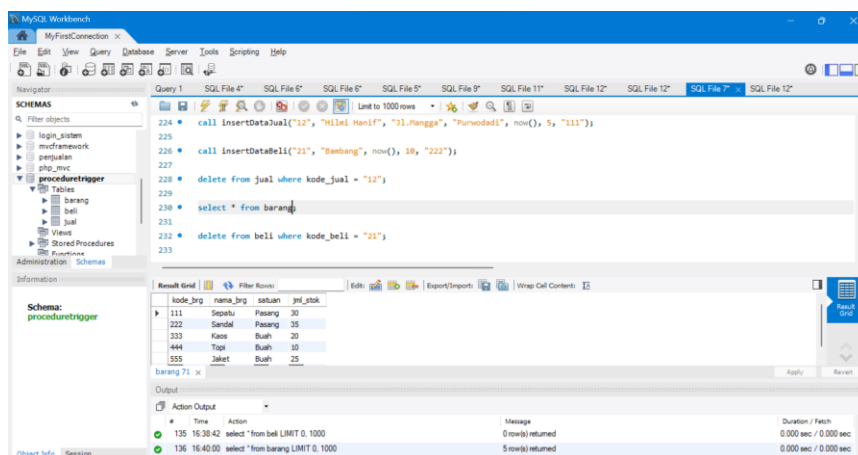
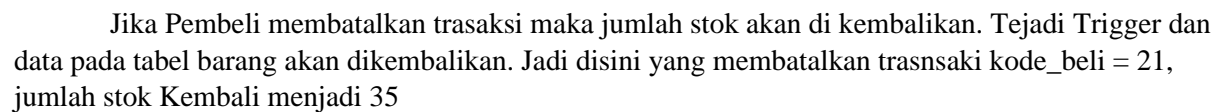


Memasukan Data Beli menggunakan store Procedure, Jika data beli dimasukan dengan jumlah beli tertentu maka data tabel barang yang memiliki hubungan kode barang maka akan di Trigger dan data pada tabel barang akan ditambah. Jadi disini jumlah barang pada kode\_brg = 222 akan ditambah menjadi 45



Jika Penjual membatalkan transaksi maka jumlah stok akan di kembalikan. Tejadi Trigger dan data pada tabel barang akan dikembalikan. Jadi disini yang membatalkan trasnsaksi kode\_jual = 12, jumlah stok Kembali menjadi 30





## **Tugas nomor 5**

- A. Apabila terjadi perubahan data jual untuk kode\_brg tertentu, maka jml\_stok disesuaikan.
- B. Apabila terjadi perubahan data beli untuk kode\_brg tertentu, maka jml\_stok disesuaikan.

### **Create Trigger**

#### **Update Jual**

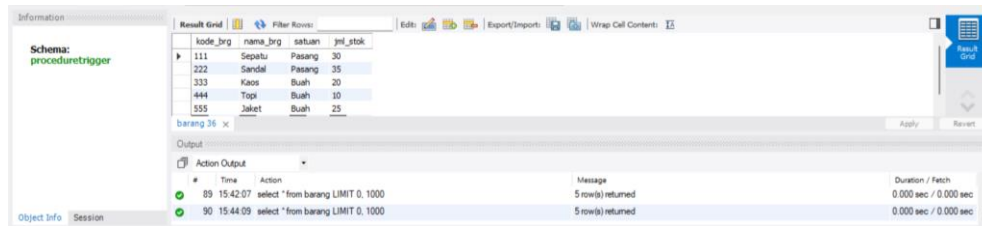
```
delimiter $$
create trigger updateJual
    after update
    on jual
    for each row
begin
    update barang
    set jml_stok = (jml_stok + old.jml_jual) - new.jml_jual
    where kode_brg = old.kode_brg;
end$$
delimiter ;
```

#### **Update Beli**

```
delimiter $$
create trigger updateBeli
    before update
    on beli
    for each row
begin
    update barang
    set jml_stok = (jml_stok - old.jml_beli) + new.jml_beli
    where kode_brg = old.kode_brg;
end$$
delimiter ;
```



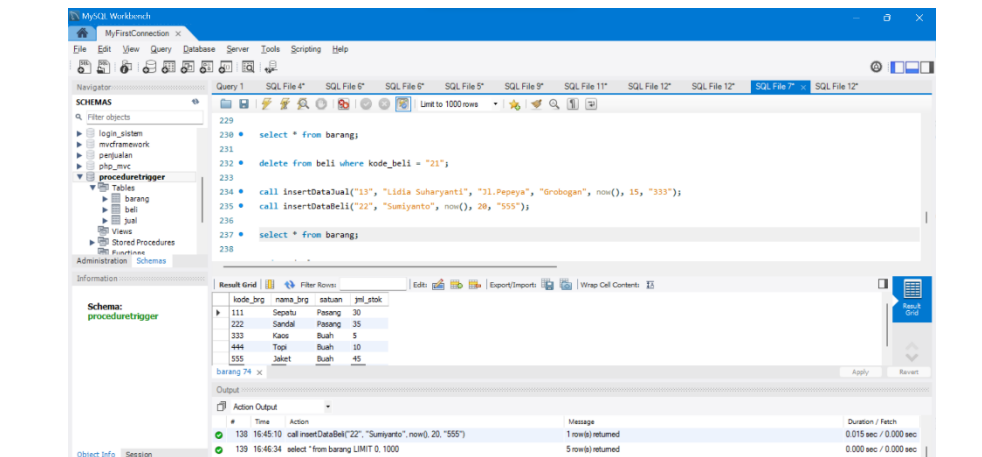
## Data Awal



| kode_brg | nama_brg | satuan | jml_stok |
|----------|----------|--------|----------|
| 111      | Sepatu   | Pasang | 30       |
| 222      | Sandal   | Pasang | 35       |
| 333      | Kaos     | Buah   | 20       |
| 444      | Topi     | Buah   | 10       |
| 555      | Jaket    | Buah   | 25       |

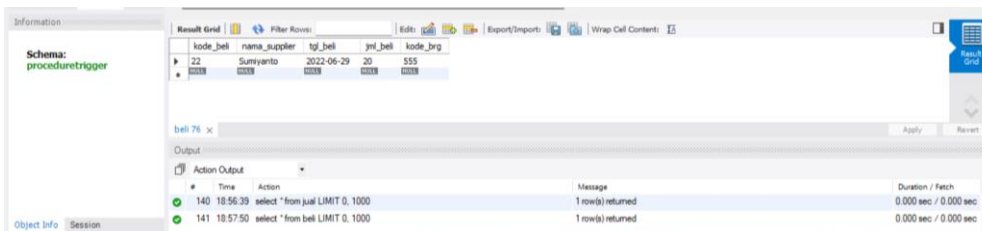
## Hasil

Saya menambahkan data terlebih dahulu pada tabel jual dan beli  
Gambaran setelah data jual dan beli di tambahkan;

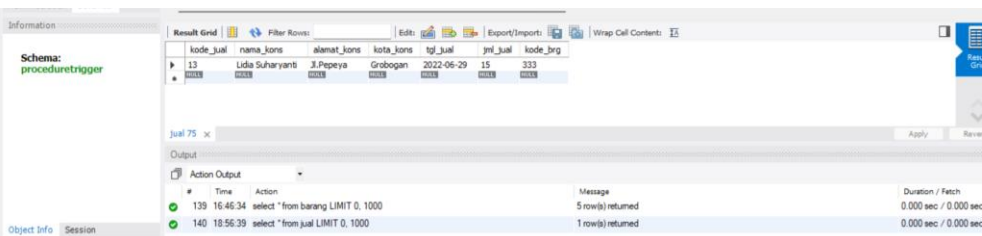


```
230 select * from barang;
231
232 delete from beli where kode_beli = "21";
233
234 call insertDataJual("13", "Lidia Suharyanti", "31.Pepeya", "Grobogan", now(), 15, "333");
235 call insertDataBeli("22", "Sumiyanto", now(), 20, "555");
236
237 select * from barang;
```

| kode_brg | nama_brg | satuan | jml_stok |
|----------|----------|--------|----------|
| 111      | Sepatu   | Pasang | 30       |
| 222      | Sandal   | Pasang | 35       |
| 333      | Kaos     | Buah   | 5        |
| 444      | Topi     | Buah   | 10       |
| 555      | Jaket    | Buah   | 45       |



| kode_beli | nama_supplier | tgl_beli   | jml_beli | kode_brg |
|-----------|---------------|------------|----------|----------|
| 22        | Sumiyanto     | 2022-06-29 | 20       | 555      |



| kode_jual | nama_kons        | alamat_kons | kota_kons | tgl_jual   | jml_jual | kode_brg |
|-----------|------------------|-------------|-----------|------------|----------|----------|
| 13        | Lidia Suharyanti | 31.Pepeya   | Grobogan  | 2022-06-29 | 15       | 333      |

Jika Update data dilakukan pada tabel jual dan beli maka data pada tabel barang akan menyesuaikan jumlah barang yang di update pada tabel jual maupun beli Karena adanya trigger

Gambaran Tabel setelah data jual diupdate

The screenshot shows the MySQL Workbench interface. The query window contains the following SQL statements:

```

232 delete from beli where kode_beli = "21";
233
234 call insertDataJual("13", "Lidia Suharyanti", "Jl.Pepeya", "Grobogan", now(), 15, "333");
235 call insertDataBeli("22", "Sumiyanto", now(), 20, "555");
236
237 update jual
238 set jml_jual = 5
239 where kode_jual = "13";
240
241 select * from barang;
    
```

The result grid shows the 'barang' table with the following data:

| kode_brg | nama_brg | satuan | jml_stok |
|----------|----------|--------|----------|
| 111      | Sepatu   | Pasang | 30       |
| 222      | Sandal   | Pasang | 35       |
| 333      | Kaos     | Buah   | 15       |
| 444      | Topi     | Buah   | 10       |
| 555      | Jaket    | Buah   | 45       |

The output window shows the execution results:

| #   | Time     | Action  | Message  | Duration / Fetch      |
|-----|----------|---|--|-----------------------|
| 145 | 19:07:22 | update jual set jml_jual = 5 where kode_jual = "13" | 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 | 0.000 sec             |
| 146 | 19:07:25 | select * from barang LIMIT 0, 1000                  | 5 row(s) returned  | 0.000 sec / 0.000 sec |

The screenshot shows the MySQL Workbench interface. The query window contains the following SQL statements:

```

13 update jual
14 set jml_jual = 5
15 where kode_jual = "13";
16
17 select * from jual;
    
```

The result grid shows the 'jual' table with the following data:

| kode_jual | nama_kons        | alamat_kons | kota_kons | tgl_jual   | jml_jual | kode_brg |
|-----------|------------------|-------------|-----------|------------|----------|----------|
| 13        | Lidia Suharyanti | Jl.Pepeya   | Grobogan  | 2022-06-29 | 5        | 333      |

The output window shows the execution results:

| #   | Time     | Action                             | Message           | Duration / Fetch      |
|-----|----------|------------------------------------|-------------------|-----------------------|
| 149 | 19:11:04 | select * from barang LIMIT 0, 1000 | 5 row(s) returned | 0.000 sec / 0.000 sec |
| 150 | 19:13:05 | select * from jual LIMIT 0, 1000   | 1 row(s) returned | 0.000 sec / 0.000 sec |

Gambaran tabel setelah data beli diupdate

The screenshot shows the MySQL Workbench interface. The query window contains the following SQL statements:

```

243
244 update jual
245 set jml_jual = 5
246 where kode_jual = "13";
247
248 update beli
249 set jml_beli = 15
250 where kode_beli = "22";
251
252 select * from barang;
    
```

The result grid shows the 'barang' table with the following data:

| kode_brg | nama_brg | satuan | jml_stok |
|----------|----------|--------|----------|
| 111      | Sepatu   | Pasang | 30       |
| 222      | Sandal   | Pasang | 35       |
| 333      | Kaos     | Buah   | 15       |
| 444      | Topi     | Buah   | 10       |
| 555      | Jaket    | Buah   | 40       |

The output window shows the execution results:

| #   | Time     | Action   | Message  | Duration / Fetch      |
|-----|----------|--|--|-----------------------|
| 216 | 19:50:19 | update beli set jml_beli = 15 where kode_beli = "22" | 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 | 0.015 sec             |
| 217 | 19:50:22 | select * from barang LIMIT 0, 1000                   | 5 row(s) returned  | 0.000 sec / 0.000 sec |

AdministrationSchemas

Information

Schema:  
proceduretrigger

Result Grid

Filter Rows

EditExport/ImportWrap Cell Content

|    | kode_bel   | nama_supplier | tgl_bel | jml_bel | kode_brg |
|----|------------|---------------|---------|---------|----------|
| 22 | Sunniyanto | 2022-06-29    | 15      | 555     |          |

beli 82 x

ApplyRevert

Output

Action Output

| #   | Time     | Action                          | Message           | Duration / Fetch      |
|-----|----------|---------------------------------|-------------------|-----------------------|
| 150 | 19:13:05 | select * from jual LIMIT 0.1000 | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 151 | 19:13:52 | select * from beli LIMIT 0.1000 | 1 row(s) returned | 0.000 sec / 0.000 sec |

Object InfoSession